LEARNING FILTERBANKS FROM RAW SPEECH FOR PHONE RECOGNITION

Neil Zeghidour[1,2], *Nicolas Usunier*[1], *Iasonas Kokkinos*[1], *Thomas Schatz*[2],
Gabriel Synnaeve[1], *Emmanuel Dupoux*[2]

[1] Facebook A.I. Research, Paris, France, New York, USA
[2] CoML, ENS/CNRS/EHESS/INRIA/PSL Research University, Paris, France

ABSTRACT

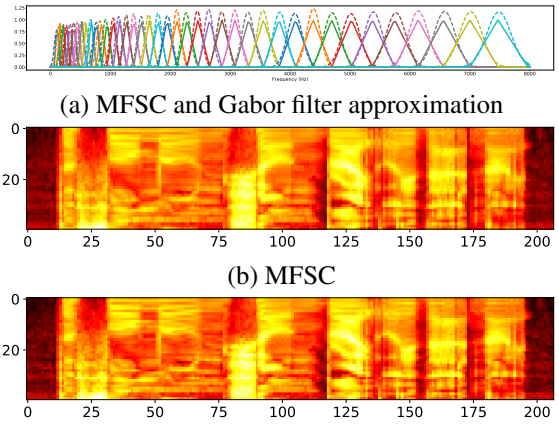
We train a bank of complex filters that operates on the raw waveform and is fed into a convolutional neural network for end-to-end phone recognition. These time-domain filterbanks (TD-filterbanks) are initialized as an approximation of mel-filterbanks, and then fine-tuned jointly with the remaining convolutional architecture. We perform phone recognition experiments on TIMIT and show that for several architectures, models trained on TD-filterbanks consistently outperform their counterparts trained on comparable mel-filterbanks. We get our best performance by learning all front-end steps, from pre-emphasis up to averaging. Finally, we observe that the filters at convergence have an asymmetric impulse response, and that some of them remain almost analytic.

1. INTRODUCTION

Speech features such as gammatones or mel-filterbanks (MFSC, for mel-frequency spectral coefficients) were designed to match the human perceptual system [1, 2], and contain invaluable priors for speech recognition tasks. However, even if a consensus has been reached on the proper setting of the hyperparameters of these filterbanks along the years, there is no reason to believe that they are optimal representations of the input signal for all recognition tasks. In the same way deep architectures changed the landscape of computer vision by directly learning from raw pixels [3, 4], we believe that future end-to-end speech recognition system will learn directly from the waveform.

There have been several attempts at learning directly from the raw waveform for speech recognition [5, 6, 7, 8]. [6, 7] propose an architecture composed of a convolutional layer followed by max-pooling and a nonlinearity, so that gammatone filterbanks correspond to a particular configuration of the network. [8] explore an alternative architecture, with the intention to represent MFSC rather than gammatones. They propose a 4-layer convolutional architecture followed by two networks-in-networks [9], pretrained to reproduce MFSC.

We also focus on MFSC because they are the front-end of state-of-the-art phone [10] and speech [11] recognition systems. Our work builds on [12], who introduce a time-domain



(a) MFSC and Gabor filter approximation

(b) MFSC

(c) TD-filterbanks with the Gabor filters

Fig. 1. Frequency response of filters, and output of MFSC and their time-domain approximation on a sentence of TIMIT.

approximation of MFSC using the first-order coefficients of a scattering transform. This leads us to study an architecture using a convolutional layer with complex-valued weights, followed by a modulus operator and a low-pass filter. In contrast to [8], we propose a lightweight architecture that serves as a plug-in, learnable replacement to MFSC in deep neural networks. Moreover, we avoid pretraining by initializing the complex convolution weights with Gabor wavelets whose center frequency and bandwidth match those of MFSC.

We perform phone recognition experiments on TIMIT and show that given competitive end-to-end models trained with MFSC as inputs, training the same architectures by replacing the MFSC with the learnable architecture leads to performances that are better than when using MFSC. Moreover, our best model is obtained by learning everything except for the non-linearities, including a pre-emphasis layer.

2. TIME-DOMAIN MFSC

We present the standard MFSC and their practical implementation. We then describe a learnable replacement of MFSC that uses only convolution operations in time domain, and how to set the weights to reproduce MFSC.

2.1. MFSC computation

Given an input signal x, MFSC are computed by first taking the short-time Fourier transform (STFT) of x followed by taking averages in the frequency domain according to triangular filters with centered frequency and bandwidth that increase linearly in log-scale. More formally, let ϕ be a Hanning window of width s and $(\psi_n)_{n=1..N}$ be N filters whose squared frequency response are triangles centered on $(\eta_n)_{n=1..N}$ with full width at half maximum (FWHM) $(w_n)_{n=1..N}$. Denoting by $x_t : u \mapsto x(u)\phi(t-u)$ the windowed signal at time step t, and $\hat{f}$ the Fourier transform of function f, the filterbank is the set of N functions $(t \mapsto Mx(t,n))_{n=1..N}$:

$$Mx(t,n) = \frac{1}{2\pi}\int |\hat{x}_t(\omega)|^2 |\hat{\psi}_n(\omega)|^2 d\omega\,, \qquad (1)$$

2.2. Approximating MFSC with convolutions in time

As in [12], we approximate MFSC in the time domain using:

$$Mx(t,n) \approx |x * \varphi_n|^2 * |\phi|^2(t). \qquad (2)$$

where φ_n is a wavelet that approximates the n-th triangular filter in frequency, i.e. $|\hat{\varphi}_n|^2 \approx |\hat{\psi}_n|^2$, while $\phi(t)$ is the Hanning window also used for the MFSC. The approximation is valid when the time support of φ_n is smaller than that of ϕ.

This approximation of MFSC is also known as a first-order scattering transform. This is the foundation of the deep scattering spectrum [12], which cascades scattering transforms to retrieve information that is lost in the MFSC. Deep scattering spectra have been used as inputs to neural networks trained for phone recognition [13] or classification [14], which showed better performances than comparable models trained on MFSC. In this work, we do not use the deep scattering spectrum. First-order scattering coefficients provide us with both a design for the first layers of the network architecture to operate on the waveform, and an initialization that approximates the MFSC computation.

Given the MFSC center frequencies $(\eta_n)_{n=1..N}$ and FWHM $(w_n)_{n=1..N}$, we use (2) to approximate MFSC with Gabor wavelets:

$$\varphi_n(t) \propto e^{-2\pi i \eta_n t} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{t^2}{2\sigma_n^2}}. \qquad (3)$$

where η_n is the desired center frequency, and the width parameter σ_n of the Gabor wavelet is set to match the desired FWHM w_n. Since for a frequency ξ we have $\hat{\varphi}_n(\xi) \propto \sqrt{\sigma_n} e^{-\frac{1}{2}\sigma_n^2(\xi-\eta_n)^2}$, the FWHM is $2\sqrt{2\log 2}\sigma_n^{-1}$ and we take $\sigma_n = \frac{2\sqrt{2\log 2}}{w_n}$. Each φ_n is then normalized to have the same energy as ψ_n. Figure 1 (a) shows in frequency-domain the triangular averaging operators of usual MFSC and the corresponding Gabor wavelets. Figures 1 (b) and (c) compare the 40-dimensional spectrograms of the MFSC and the Gabor wavelet approximation on a random sentence of the TIMIT corpus after mean-variance normalization, showing that the spectrograms are similar.

Layer type	Input size	Output size	Width	Stride
Conv.	1	80	400	1
L2-Pooling	80	40	-	-
Square	-	-	-	-
Grouped conv.	40	40	400	160
Absolute value	-	-	-	-
Add 1, Log	-	-	-	-

Table 1. Details of the layers for the TD-filterbanks.

Learning mode	Dev PER	Test PER
MFSC	17.8	20.6
Fixed	18.3	21.8
Learn-all	17.4	20.6
Learn-filterbank	17.3	20.3
Randinit	29.2	31.7

Table 2. PER of the CNN-5L-ReLU-do0.7 model trained on MFSC and different learning setups of TD-filterbanks.

MFSC specification. The standard setting in speech recognition is to start from the waveform sampled at 16kHz and represented as 16-bit signed integers. The STFT is computed with 512 frequency bins using Hanning windows of width 25ms, and decimation is applied by taking the STFT every 10ms. There are $N = 40$ filters, with center frequencies $(\eta_n)_{n=1..N}$ that span the range $64Hz - 8000Hz$ by being equally spaced on a mel-scale. The final features are the $\log(\max(Mx(t,n),1))$. In practice, the STFT is applied to the raw signal after a pre-emphasis with parameter 0.97, and coefficients have mean-variance normalization per utterance.

Learnable architecture specification. The time-domain convolutional architecture is summarized in Table 1. With a waveform sampled at 16kHz, a Hanning window is a convolution operator with a span of $W = 400$ samples (25ms). Since the energy of the Gabor wavelets approximating standard MFSC has a time spread smaller than the Hanning window, the complex wavelet+modulus operations $|x * \varphi_n|^2$ are implemented as a convolutional layer taking the raw wav as input, with a width $W = 400$ and $2N = 80$ filters (40 filters for the real and imaginary parts respectively). This layer is on the top row of Table 1. The modulus operator is implemented with "feature L2 pooling", a layer taking an input z of size $2N$ and outputs z' of size N such that $z'_k = \sqrt{z_{2k-1}^2 + z_{2k}^2}$. The windowing layer (third row of Table 1) is a grouped convolution, meaning that each output filter only sees the input filter with the same index. The decimation of 10ms is implemented in the stride of 160 of this layer. Notice that to approximate the mel-filterbanks, the square of the Hanning window is used and biases in both convolutional layers are set to zero. We keep them to zero during training. We add log compression to the output of the grouped convolution after adding 1 to its absolute value since we do not have positivity constraints on

Model	Input	Dev PER	Test PER
Hybrid HMM/Hierarchical CNN + Maxout + Dropout [10]	MFSC + energy + Δ + $\Delta\Delta$	13.3	16.5
CNN + CRF on raw speech [15]	wav	-	29.2
Wavenet [16]	wav	-	18.8
CNN-Conv2D-10L-Maxout [17]	MFSC	16.7	18.2
Attention model + Conv. Features + Smooth Focus [18]	MFSC + energy + Δ + $\Delta\Delta$	15.8	17.6
LSTM + Segmental CRF [19]	MFSC + Δ + $\Delta\Delta$	-	18.9
LSTM + Segmental CRF [19]	MFCC + LDA + MLLT + MLLR	-	17.3
CNN-5L-ReLU-do0.5	MFSC	18.4	20.8
CNN-5L-ReLU-do0.5 + TD-filterbanks	wav	18.2	20.4
CNN-5L-ReLU-do0.7	MFSC	17.8	20.6
CNN-5L-ReLU-do0.7 + TD-filterbanks	wav	17.3	20.3
CNN-8L-PReLU-do0.7	MFSC	16.2	18.1
CNN-8L-PReLU-do0.7 + TD-filterbanks	wav	15.6	18.1
CNN-8L-PReLU-do0.7 + TD-filterbanks-Learn-all-pre-emp	wav	15.6	18.0

Table 3. PER (Phone Error Rate) on TIMIT, in percentages. All models but [10] are trained in an end-to-end fashion.

the weights when learning. Contrarily to the MFSC, there is no mean-variance normalization after the convolutions, but on the waveform. In the default implementation of the TD-filterbanks, we do not apply pre-emphasis. However, in our last experiment, we add a convolutional layer below the TD-filterbanks, with width 2 and stride 1, initialized with the pre-emphasis parameters, as another learnable component.

3. EXPERIMENTS

3.1. Setting

We perform phone recognition experiments on TIMIT [20] using the standard train/dev/test split. We train and evaluate our models with 39 phonemes. We experiment with three architectures. The first one consists of 5 layers of convolution of width 5 and 1000 feature maps, with ReLU activation functions, and a dropout [21] of 0.5 on every layer but the input and output ones. The second model has the same architecture but a dropout of 0.7 is used. The third model has 8 layers of convolution, PReLU [22] nonlinearities and a dropout of 0.7. All our models are trained end-to-end with the Autoseg criterion [23], using stochastic gradient descent. We compare all models using either the baseline MFSC as input or our learnable TD-filterbank front-end. We perform the same grid-search for both MFSC baselines and models trained on TD-filterbanks, using learning rates in $(0.0003, 0.003)$ for the model and learning rates in $(0.03, 0.003)$ for the Autoseg criterion, training every model for 2000 epochs. We use the standard dev set for early stopping and hyperparameter selection.

3.2. Different types of TD-filterbanks

Throughout our experiments, we tried four different settings for the TD-filterbank layers:

- Fixed: Initialize the layers to match MFSC and keep their parameters fixed when training the model
- Learn-all: Initialize the layers and let the filterbank and the averaging be learned jointly with the model
- Learn-filterbank: Start from the initialization and only learn the filterbank with the model, keeping the averaging fixed to a squared hanning window
- Randinit: Initialize the layers randomly and learn them with the network

Table 2 shows comparative performance of an identical architecture trained on the four types of TD-filterbanks. We can observe that training on fixed layers moderately worsens the performance, we hypothesize that this is due to the absence of mean-variance normalization on top of TD-filterbanks as is performed on MFSC. A striking observation is that a model trained on TD-filterbanks initialized randomly performs considerably worse than all other models. This shows the importance of the initialization. Finally, we observe better results when learning the filterbank only compared to learning the filterbank and the averaging but depending on the architecture it was not clear which one performs better. Moreover, when learning both complex filters and averaging, we observe that the learned averaging filters are almost identical to their initialization. Thus, in the following experiments, we choose to use the Learn-filterbank mode for the TD-filterbanks.

3.3. Results

We report PER on the standard dev and test sets of TIMIT. For each architecture, we can observe that the model trained on TD-filterbanks systematically outperforms the equivalent model trained on MFSC, even though we constrained our TD-filterbanks such that they are comparable to the MFSC and do not learn the low-pass filter. This shows that by only learning a new bank of 40 filters, we can outperform the MFSC for phone recognition. This gain in performance is obtained at a minimal cost in terms of number of parameters: even for the smallest architecture, the increase in number of parameters

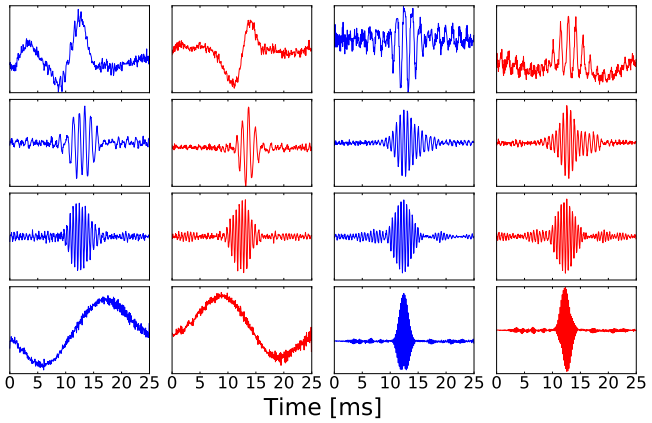


Fig. 2. Examples of learned filters. Filters' real parts in blue; imaginary part in red.

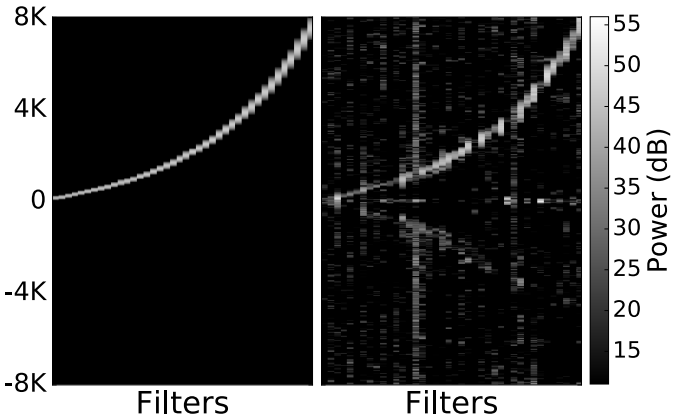


Fig. 3. Heat-map of the magnitude of the frequency response for initialization filters (left) and learned filters (right).

in switching from MFSC to TD-filterbanks is 0.31%. We also compare to baselines from the literature. One baseline trained on the waveform gets a PER of 29.1% on the test set, which is in a range $8.8\% - 11.1\%$ absolute above our models trained on the waveform. The Wavenet architecture, also trained on the waveform, yields a PER of 18.8, which is higher than our best models despite using the phonetic alignment and an auxiliary prediction loss. Our best model on the waveform also outperforms a 2-dimensional CNN trained on MFSC and an LSTM trained on MFSC with derivatives. Finally, by adding a learnable pre-emphasis layer below the TD-filterbanks, we reach 18% PER on the test set.

4. ANALYSIS OF LEARNED FILTERS

We analyze filters learned by the first layer of the CNN-8L-PReLU-do0.7 + TD-filterbanks model. Examples of learned filters are shown in Figure 2. The magnitude of the frequency response for each of the 40 filters is plotted in Figure 3. Overall, the filters tend to be well localized in time and frequency, and a number of filters became asymmetric during the learning process, with a sharp attack and slow decay of the impulse response. This is a type of asymmetry also found in human and animal auditory filters estimated from behavioral and physiological data [24]. In Figure 3, we further see that the initial mel-scale of frequency is mostly preserved, but that a lot of variability in the filter bandwidths is introduced.

A prominent question is whether the analyticity of the initial filterbank is preserved throughout the learning process even though nothing in our optimization method is biased towards keeping filters analytic. A positive answer would suggest that complex filters in their full generality are not necessary to obtain the increase in performance we observed. This would be especially interesting because, unlike arbitrary complex filters, analytic filters have a simple interpretation in terms of real-domain signal processing: taking the squared modulus of the convolution of a real signal with an analytic filter performs a sub-band Hilbert envelope extraction [25].

A signal is analytic if and only if it has no energy in the negative frequencies. Accordingly, we see in Figure 3 that there is zero energy in this region for the initialization filterbank. After learning, a moderate amount of energy appears in the negative frequency region for certain filters. To quantify this, we computed for each filter the ratio r_a between the energy in negative versus positive frequency components [1]. This ratio is 0 for a perfectly analytic filter and 1 for a purely real filter. We find an average r_a for all learned filters of .26. Filters with significant energy in negative frequencies are mostly filters with an intermediate preferred frequency (between 1000Hz and 3000Hz) and their negative frequency spectrum appears to be essentially a down-scaled version of their positive frequency spectrum.

5. CONCLUSION

We proposed a lightweight architecture which, at initialization, approximates the computation of MFSC and can then be fine-tuned with an end-to-end phone recognition system. With a number of parameters comparable to standard MFSC, a TD-filterbank front-end is consistently better in our experiments. Learning all linear operations in the MFSC derivation, from pre-emphasis up-to averaging provides the best model. In future work, we will perform large scale experiments with TD-filterbanks to test if a new state-of-the-art can be achieved by training from the waveform.

6. ACKNOWLEDGEMENTS

Authors thank Mark Tygert for useful discussions, and Vitaliy Liptchinsky and Ronan Collobert for help on the implementation. This research was partially funded by the European Research Council (ERC-2011-AdG-295810 BOOTPHON), the Agence Nationale pour la Recherche (ANR-10-LABX-0087 IEC, ANR-10-IDEX-0001-02 PSL*).

[1] Our model cannot identify if a given filter plays the role of the real or imaginary part in the associated complex filter. We chose the assignment yielding the smallest r_a.

7. REFERENCES

[1] Steven Davis and Paul Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[2] RD Patterson, Ian Nimmo-Smith, John Holdsworth, and Peter Rice, "An efficient auditory filterbank based on the gammatone function," in *a meeting of the IOC Speech Group on Auditory Modelling at RSRE*, 1987, vol. 2.

[3] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[5] Dimitri Palaz, Ronan Collobert, and Mathew Magimai Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," *arXiv preprint arXiv:1304.1018*, 2013.

[6] Yedid Hoshen, Ron J Weiss, and Kevin W Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Proceedings of ICASSP*. IEEE, 2015.

[7] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Interspeech*, 2015.

[8] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Attention-based wav2text with feature transfer learning," *arXiv preprint arXiv:1709.07814*, 2017.

[9] Min Lin, Qiang Chen, and Shuicheng Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[10] László Tóth, "Phone recognition with hierarchical convolutional deep maxout networks," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, pp. 25, 2015.

[11] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[12] Joakim Andén and Stéphane Mallat, "Deep scattering spectrum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, 2014.

[13] Vijayaditya Peddinti, TaraN Sainath, Shay Maymon, Bhuvana Ramabhadran, David Nahamoo, and Vaibhava Goel, "Deep scattering spectrum with deep neural networks," in *ICASSP*. IEEE, 2014.

[14] Neil Zeghidour, Gabriel Synnaeve, Maarten Versteegh, and Emmanuel Dupoux, "A deep scattering spectrumdeep siamese network pipeline for unsupervised acoustic modeling," in *ICASSP*. IEEE, 2016.

[15] Dimitri Palaz, Ronan Collobert, and Mathew Magimai Doss, "End-to-end phoneme sequence recognition using convolutional neural networks," *arXiv preprint arXiv:1312.2137*, 2013.

[16] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[17] Ying Zhang, Mohammad Pezeshki, Philémon Brakel, Saizheng Zhang, Cesar Laurent Yoshua Bengio, and Aaron Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," *arXiv preprint arXiv:1701.02720*, 2017.

[18] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *NIPS*.

[19] Liang Lu, Lingpeng Kong, Chris Dyer, Noah A Smith, and Steve Renals, "Segmental recurrent neural networks for end-to-end speech recognition," *arXiv preprint arXiv:1603.00223*, 2016.

[20] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, David S Pallett, Nancy L Dahlgren, and Victor Zue, "Timit acoustic-phonetic continuous speech corpus," *Linguistic data consortium*, vol. 10, no. 5, pp. 0, 1993.

[21] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *CVPR*, 2015.

[23] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[24] Evan C Smith and Michael S Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 7079, pp. 978–982, 2006.

[25] JL Flanagan, "Parametric coding of speech spectra," *The Journal of the Acoustical Society of America*, vol. 68, no. 2, pp. 412–419, 1980.